

Project

Gonçalo Silva

David Sousa

Versão 1.1

Domingo, 24 de Janeiro de 2016

Índice

Table of contents

Índice dos namespaces

Lista de namespaces

Lista dos namespaces com uma breve descrição:

webservice	5
WS1	6

Índice das estruturas de dados

Estruturas de dados

Lista das estruturas de dados com uma breve descrição:

condutor	7
condutor_veiculo	9
localizacao	11
Percurso	13
percurso_ponto	15
php	17
Ponto	18
veiculo	20
WS1_webservice	23

Índice dos ficheiros

Lista de ficheiros

Lista de todos os ficheiros com uma breve descrição:

class.condutor.php	28
class.condutor_veiculo.php	29
class.localizacao.php	30
class.percurso.php	31
class.percurso_ponto.php	32
class.ponto.php	33
class.veiculo.php	34
class.webservice.php	35
index.php	36
interface.php	37
interface_condutor.php	38
interface_condutor_veiculo.php	39
interface_localizacao.php	40
interface_percurso.php	41
interface_percurso_ponto.php	42
interface_ponto.php	43
interface_veiculo.php	44

Documentação dos namespaces

Referência ao namespace webservice

Estruturas de Dados

- class `php`

Referência ao namespace WS1

Descrição detalhada

include WS1_user

Autor:

firstname and lastname of author, author@example.org include nusoap library

firstname and lastname of author, author@example.org include our library with some DB functions

firstname and lastname of author, author@example.org Short description of class **WS1_webservice**

public

Autor:

firstname and lastname of author, author@example.org

Documentação da classe

Referência à classe condutor

Membros públicos

- `__construct ($nome, $bi, $carta_conducao)`
- `getNome ()`
- `getBI ()`
- `getCarta_Conducao ()`
- `criar_condutor ()`

Membros públicos estáticos

- `static query_all_condutor ()`

Campos de Dados

- `$nome = null`
- `$bi = 0`
- `$carta_conducao = 0`

Descrição detalhada

include veiculo_condutor

Autor:

firstname and lastname of author, author@example.org include veiculo_condutor_percurso

firstname and lastname of author, author@example.org Short description of class condutor

public

Autor:

firstname and lastname of author, author@example.org

Documentação dos Construtores & Destrutor

`__construct ($nome, $bi, $carta_conducao)`

```
63      {
64          $this->nome = $nome;
65          $this->bi = $bi;
66          $this->carta_conducao = $carta_conducao;
67      }
```

Documentação dos métodos

`criar_condutor ()`

```
83      {
84          $query = "insert into condutor (nome,bi,carta_conducao) values (?, ?, ?)";
85
86          $result = queryMysqli($query,
array($this->getNome(), $this->getBI(), $this->getCarta_Conducao()));
```

```

87
88     if ($result->rowCount() > 0) {
89         return true;
90     }
91     return false;
92 }

```

getBI ()

```

74     {
75         return $this->bi;
76     }

```

getCarta_Conducao ()

```

78     {
79         return $this->carta_conducao;
80     }

```

getNome ()

```

70     {
71         return $this->nome;
72     }

```

static query_all_condutor () [static]

```

94     {
95         $query = "select * from condutor";
96         $queryResult=array();
97         $result = queryMysql($query, array());
98         if ($result->rowCount() <= 0) {
99             return null;
100         } else {
101             for ($i = 0; $i < $result->rowCount(); $i++) {
102                 $object = $result->fetchObject();
103                 array_push($queryResult, $object->nome . ' ' . $object->bi . ' ' .
$object->carta_conducao . ' ');
104             }
105         }
106         return $queryResult;
107     }

```

Documentação dos campos e atributos

\$bi = 0

\$carta_conducao = 0

\$nome = null

A documentação para esta classe foi gerada a partir do seguinte ficheiro:

- **class.condutor.php**

Referência à classe condutor_veiculo

Membros públicos

- `__construct ($id_condutor, $id_veiculo)`
- `getIdCondutor ()`
- `getIdVeiculo ()`
- `criar_condutor_veiculo ()`

Membros públicos estáticos

- `static query_all_condutor_veiculo ()`

Campos de Dados

- `$id = null`
- `$id_condutor = 0`
- `$id_veiculo = 0`

Descrição detalhada

include percurso_ponto

Autor:

firstname and lastname of author, author@example.org Short description of class **Ponto**

public

Autor:

firstname and lastname of author, author@example.org

Documentação dos Construtores & Destrutor

`__construct ($id_condutor, $id_veiculo)`

```
53                                     {
54         $this->id_condutor = $id_condutor;
55         $this->id_veiculo = $id_veiculo;
56     }
```

Documentação dos métodos

`criar_condutor_veiculo ()`

```
67     {
68         $query = "insert into condutor_veiculo (id_condutor,id_veiculo) values
69         (?,?)";
69
70         $result = queryMysql($query,
71         array($this->getIdCondutor(),$this->getIdVeiculo()));
71
72         if ($result->rowCount() > 0) {
73             return true;
74         }
75         return false;
76     }
```

getIdCondutor ()

```
58      {
59      return $this->id_condutor;
60      }
```

getIdVeiculo ()

```
62      {
63      return $this->id_veiculo;
64      }
```

static query_all_condutor_veiculo () [static]

```
78      {
79      $query = "select condutor_veiculo.id,c.nome,v.marca,v.matricula from
condutor_veiculo INNER JOIN condutor as c INNER JOIN veiculo as v ON c.id=condutor_veiculo.id
and v.id=v.id=condutor_veiculo.id";
80      $queryResult=array();
81      $result = queryMysqli($query, array());
82      if ($result->rowCount() <= 0) {
83          return null;
84      } else {
85          for ($i = 0; $i < $result->rowCount(); $i++) {
86              $object = $result->fetchObject();
87              array_push($queryResult, $object->id . ' ' . $object->nome . '
' . $object->marca . ' ' . $object->matricula . ' ');
88          }
89      }
90      return $queryResult;
91      }
```

Documentação dos campos e atributos

\$id = null

\$id_condutor = 0

\$id_veiculo = 0

A documentação para esta classe foi gerada a partir do seguinte ficheiro:

- class.condutor_veiculo.php

Referência à classe localizacao

Membros públicos

- `__construct ($latitude, $longitude)`
- `getLatitude ()`
- `getLongitude ()`
- `criar_localizacao ()`

Membros públicos estáticos

- `static update_localizacao ($lat1, $lon1, $id_localizacao1)`
- `static query_all_localizacao ()`

Campos de Dados

- `$latitude = 0.0`
- `$longitude = 0.0`
- `$id = 0`

Descrição detalhada

Short description of class localizacao

public

Autor:

firstname and lastname of author, author@example.org

Documentação dos Construtores & Destrutor

`__construct ($latitude, $longitude)`

```
53                                     {
54         $this->latitude = $latitude;
55         $this->longitude = $longitude;
56     }
```

Documentação dos métodos

`criar_localizacao ()`

```
67     {
68         $query = "insert into localizacao (latitude,longitude) values (?,?)";
69
70         $result = queryMysql($query,
array($this->getLatitude(), $this->getLongitude()));
71
72         if ($result->rowCount() > 0) {
73             return true;
74         }
75         return false;
76     }
```

`getLatitude ()`

```
58                                     {
```

```

59         return $this->latitude;
60     }

```

getLongitude ()

```

62         {
63         return $this->longitude;
64     }

```

static query_all_localizacao ()[static]

```

89         {
90         $query = "select * from localizacao";
91         $queryResult=array();
92         $result = queryMysqli($query, array());
93         if ($result->rowCount() <= 0) {
94             return null;
95         } else {
96             for ($i = 0; $i < $result->rowCount(); $i++) {
97                 $object = $result->fetchObject();
98                 array_push($queryResult, $object->latitude . ' ' . $object->longitude
99                 . ' ');
100             }
101             return $queryResult;
102         }

```

static update_localizacao (\$lat1, \$lon1, \$id_localizacao1)[static]

```

79         {
80         $query = "update localizacao set latitude=".$lat1.",longitude=".$lon1." where
id=".$id_localizacao1;
81
82         $result = queryMysqli($query, array());
83
84
85         return $result;
86
87     }

```

Documentação dos campos e atributos

\$id = 0

\$latitude = 0.0

\$longitude = 0.0

A documentação para esta classe foi gerada a partir do seguinte ficheiro:

- class.localizacao.php

Referência à classe Percurso

Membros públicos

- `__construct ($descricao)`
- `getDescricao ()`
- `criar_percurso ()`

Membros públicos estáticos

- `static query_all_percurso ()`

Campos de Dados

- `$descricao = null`
- `$id = 0`

Descrição detalhada

include `percurso_ponto`

Autor:

firstname and lastname of author, `author@example.org` Short description of class **Ponto**

public

Autor:

firstname and lastname of author, `author@example.org`

Documentação dos Construtores & Destrutor

`__construct ($descricao)`

```
52      {
53          $this->descricao = $descricao;
54      }
```

Documentação dos métodos

`criar_percurso ()`

```
61      {
62          $query = "insert into percurso (descricao) values (?)";
63          $result = queryMysqli($query, array($this->getDescricao()));
64          if ($result->rowCount() > 0) {
65              return true;
66          }
67          return false;
68      }
```

`getDescricao ()`

```
56      {
57          return $this->descricao;
58      }
```


static query_all_percurso () [static]

```
72                                     {
73     $query = "select * from percurso";
74     $queryResult=array();
75     $result = queryMysqli($query, array());
76     if ($result->rowCount() <= 0) {
77         return null;
78     } else {
79         for ($i = 0; $i < $result->rowCount(); $i++) {
80             $object = $result->fetchObject();
81             array_push($queryResult, $object->id . ' ' . $object->descricao . '
82         ');
83     }
84     return $queryResult;
85 }
```

Documentação dos campos e atributos

\$descricao = null

\$id = 0

A documentação para esta classe foi gerada a partir do seguinte ficheiro:

- class.percurso.php

Referência à classe percurso_ponto

Membros públicos

- `__construct ($id_percurso, $id_ponto)`
- `getIdPercurso ()`
- `getIdPonto ()`
- `criar_percurso_ponto ()`

Membros públicos estáticos

- `static query_all_percurso_ponto ()`

Campos de Dados

- `$id = null`
- `$id_percurso = 0`
- `$id_ponto = 0`

Descrição detalhada

include `percurso_ponto`

Autor:

firstname and lastname of author, `author@example.org` Short description of class **Ponto**
public

Autor:

firstname and lastname of author, `author@example.org`

Documentação dos Construtores & Destrutor

`__construct ($id_percurso, $id_ponto)`

```
53      {  
54          $this->id_percurso = $id_percurso;  
55          $this->id_ponto = $id_ponto;  
56      }
```

Documentação dos métodos

`criar_percurso_ponto ()`

```
67      {  
68          $query = "insert into percurso_ponto (id_percurso,id_ponto) values (?,?)";  
69  
70          $result = queryMysql($query,  
array($this->getIdPercurso(),$this->getIdPonto()));  
71  
72          if ($result->rowCount() > 0) {  
73              return true;  
74          }  
75          return false;  
76      }
```

getIdPercurso ()

```
58      {  
59      return $this->id_percurso;  
60      }
```

getIdPonto ()

```
62      {  
63      return $this->id_ponto;  
64      }
```

static query_all_percurso_ponto () [static]

```
78      {  
79      $query = "select percurso_ponto.id,l.descricao as pontodescricao,p.descricao  
as percursodescricao from percurso_ponto INNER JOIN Percurso p ON  
p.id=percurso_ponto.id_percurso INNER JOIN Ponto l ON l.id=percurso_ponto.id_ponto";  
80      $queryResult=array();  
81      $result = queryMysqli($query, array());  
82      if ($result->rowCount() <= 0) {  
83          return null;  
84      } else {  
85          for ($i = 0; $i < $result->rowCount(); $i++) {  
86              $object = $result->fetchObject();  
87              array_push($queryResult, $object->id . ' ' . $object->pontodescricao  
. ' ' . $object->percursodescricao);  
88          }  
89      }  
90      return $queryResult;  
91      }
```

Documentação dos campos e atributos

\$id = null

\$id_percurso = 0

\$id_ponto = 0

A documentação para esta classe foi gerada a partir do seguinte ficheiro:

- class.percurso_ponto.php

Referência à classe php

Descrição detalhada

SE-Project - WS1

\$Id\$

This file is part of SE-Project.

Automatically generated on 13.12.2014, 13:01:03 with ArgoUML PHP module (last revised

Date

2010-01-12 20:14:42 +0100 (Tue, 12 Jan 2010)

)

Autor:

firstname and lastname of author, author@example.org

A documentação para esta classe foi gerada a partir do seguinte ficheiro:

- **class.webservice.php**

Referência à classe Ponto

Membros públicos

- `__construct ($descricao)`
- `getDescricao ()`
- `criar_ponto ()`

Membros públicos estáticos

- `static query_all_ponto ()`

Campos de Dados

- `$descricao = null`
- `$id = 0`

Descrição detalhada

include `percurso_ponto`

Autor:

firstname and lastname of author, `author@example.org` Short description of class **Ponto**

public

Autor:

firstname and lastname of author, `author@example.org`

Documentação dos Construtores & Destrutor

`__construct ($descricao)`

```
52      {
53          $this->descricao = $descricao;
54      }
```

Documentação dos métodos

`criar_ponto ()`

```
61      {
62          $query = "insert into ponto (descricao) values (?)";
63
64          $result = queryMysqli($query, array($this->getDescricao()));
65
66          if ($result->rowCount() > 0) {
67              return true;
68          }
69          return false;
70      }
```

`getDescricao ()`

```
56      {
57          return $this->descricao;
58      }
```

static query_all_ponto () [static]

```
72                                     {
73     $query = "select * from ponto";
74     $queryResult=array();
75     $result = queryMysqli($query, array());
76     if ($result->rowCount() <= 0) {
77         return null;
78     } else {
79         for ($i = 0; $i < $result->rowCount(); $i++) {
80             $object = $result->fetchObject();
81             array_push($queryResult, $object->id . ' ' . $object->descricao . '
82         ');
83     }
84     return $queryResult;
85 }
```

Documentação dos campos e atributos

\$descricao = null

\$id = 0

A documentação para esta classe foi gerada a partir do seguinte ficheiro:

- class.ponto.php

Referência à classe veiculo

Membros públicos

- **__construct** (\$marca, \$modelo, \$matricula, \$capacidade, \$autonomia, \$estado)
- **getMarca** ()
- **getModelo** ()
- **getMatricula** ()
- **getCapacidade** ()
- **getAutonomia** ()
- **getEstado** ()
- **criar_veiculo** ()

Membros públicos estáticos

- **static query_all_veiculo** ()

Campos de Dados

- **\$marca** = null
- **\$modelo** = null
- **\$matricula** = null
- **\$capacidade** = 0
- **\$autonomia** = 0.0
- **\$estado** = null

Descrição detalhada

include veiculo_condutor

Autor:

firstname and lastname of author, author@example.org include veiculo_condutor_percurso

firstname and lastname of author, author@example.org include veiculo_localizacao

firstname and lastname of author, author@example.org Short description of class veiculo

public

Autor:

firstname and lastname of author, author@example.org

Documentação dos Construtores & Destrutor

__construct (**\$marca**, **\$modelo**, **\$matricula**, **\$capacidade**, **\$autonomia**, **\$estado**)

```
82
{
83     $this->marca = $marca;
84     $this->modelo = $modelo;
85     $this->matricula = $matricula;
86     $this->capacidade = $capacidade;
87     $this->autonomia = $autonomia;
88     $this->estado = $estado;
89 }
```

Documentação dos métodos

criar_veiculo ()

```
130     {
131         $query = "insert into veiculo
(marca,modelo,matricula,capacidade,autonomia,estado) values (?,?,?,?,?,?)";
132
133         $result = queryMysqli($query,
array($this->getMarca(),$this->getModelo(),$this->getMatricula(),$this->getCapacidade(),
$this->getAutonomia(),$this->getEstado()));
134
135         if ($result->rowCount() > 0) {
136             return true;
137         }
138         return false;
139     }
```

getAutonomia ()

```
110     {
111         return $this->autonomia;
112     }
```

getCapacidade ()

```
107     {
108         return $this->capacidade;
109     }
```

getEstado ()

```
114     {
115         return $this->estado;
116     }
```

getMarca ()

```
95     {
96         return $this->marca;
97     }
```

getMatricula ()

```
103     {
104         return $this->matricula;
105     }
```

getModelo ()

```
99     {
100         return $this->modelo;
101     }
```

static query_all_veiculo () [static]

```
141     {
142         $query = "select * from veiculo";
143         $queryResult=array();
144         $result = queryMysqli($query, array());
145         if ($result->rowCount() <= 0) {
146             return null;
147         } else {
148             for ($i = 0; $i < $result->rowCount(); $i++) {
149                 $object = $result->fetchObject();
150                 array_push($queryResult, $object->marca . ' ' . $object->modelo . '
'. $object->matricula . ' ' . $object->capacidade . ' ' . $object->autonomia . ' ' .
$object->estado . ' ');
151             }
152         }
153         return $queryResult;
154     }
```

Documentação dos campos e atributos

\$autonomia = 0.0

\$capacidade = 0

\$estado = null

\$marca = null

\$matricula = null

\$modelo = null

A documentação para esta classe foi gerada a partir do seguinte ficheiro:

- `class.veiculo.php`

Referência à classe WS1_webservice

Membros públicos

- `__construct ()`
- `criar_veiculo ($marca, $modelo, $matricula, $capacidade, $autonomia, $estado)`
- `criar_condutor ($nome, $bi, $carta_conducao)`
- `query_all_condutor ()`
- `criar_ponto ($descricao)`
- `criar_percurso ($descricao)`
- `criar_condutor_veiculo ($id_condutor, $id_veiculo)`
- `criar_percurso_ponto ($id_percurso, $id_ponto)`
- `update_localizacao ($id, $lat, $lon)`
- `query_all_percurso_ponto ()`
- `query_all_condutor_veiculo ()`
- `query_all_ponto ()`
- `query_all_percurso ()`
- `query_all_veiculo ()`
- `query_all_localizacao ()`
- `criar_localizacao ($latitude, $longitude)`
- `registerMethod ($nameMethod)`
- `processRequest ()`

Campos de Dados

- `$server = NULL`

Documentação dos Construtores & Destrutor

`__construct ()`

```
68                                     {
69         // Create the server instance
70         $this->server = new nusoap_server();
71         // Define the method as a PHP function
72     }
```

Documentação dos métodos

`criar_condutor ($nome, $bi, $carta_conducao)`

```
114                                     {
115
116         $condutor = new condutor($nome, $bi, $carta_conducao);
117         $result = $condutor->criar_condutor();
118         if ($result) {
119             return "condutor successfully inserted";
120         }
121         return "condutor could not be inserted";
122     }
```

`criar_condutor_veiculo ($id_condutor, $id_veiculo)`

```
156                                     {
157
158         $condutor_veiculo = new condutor_veiculo($id_condutor,$id_veiculo);
159         $result = $condutor_veiculo->criar_condutor_veiculo();
```

```

160         if ($result) {
161             return "criar_condutor_veiculo successfully inserted";
162         }
163         return "criar_condutor_veiculo could not be inserted";
164     }

```

criar_localizacao (\$latitude, \$longitude)

```

252                                     {
253
254         $localizacao = new localizacao($latitude,$longitude);
255         $result = $localizacao->criar_localizacao();
256         if ($result) {
257             return "localizacao successfully inserted";
258         }
259         return "localizacao could not be inserted";
260     }

```

criar_percurso (\$descricao)

```

145                                     {
146
147         $percurso = new percurso($descricao);
148         $result = $percurso->criar_percurso();
149         if ($result) {
150             return "percurso successfully inserted";
151         }
152         return "percurso could not be inserted";
153     }

```

criar_percurso_ponto (\$id_percurso, \$id_ponto)

```

166                                     {
167
168         $percurso_ponto = new percurso_ponto($id_percurso,$id_ponto);
169         $result = $percurso_ponto->criar_percurso_ponto();
170         if ($result) {
171             return "criar percurso_ponto successfully inserted";
172         }
173         return "criar percurso_ponto could not be inserted";
174     }

```

criar_ponto (\$descricao)

```

135                                     {
136
137         $ponto = new ponto($descricao);
138         $result = $ponto->criar_ponto();
139         if ($result) {
140             return "ponto successfully inserted";
141         }
142         return "ponto could not be inserted";
143     }

```

criar_veiculo (\$marca, \$modelo, \$matricula, \$capacidade, \$autonomia, \$estado)

create an WS1_user object and make register itself. will return either a token or a message of error

public

Autor:

firstname and lastname of author, author@example.org

Parâmetros:

<i>String</i>	firstname
<i>String</i>	lastname
<i>String</i>	username

<i>String</i>	birthdate
<i>char</i>	gender

Retorna:

WS1_xml create an WS1_activity after check if the user exists, and the given token is valid
public

Autor:

firstname and lastname of author, author@example.org

Parâmetros:

<i>String</i>	username
<i>String</i>	start
<i>String</i>	end
<i>String</i>	activityName
<i>String</i>	token

Retorna:

mixed

```

104
{
105
106     $veiculo = new veiculo($marca, $modelo, $matricula, $capacidade, $autonomia,
$estado);
107     $result = $veiculo->criar_veiculo();
108     if ($result) {
109         return "veiculo successfully inserted";
110     }
111     return "veiculo could not be inserted";
112 }
```

processRequest ()

```

267                                     {
268     $this->server->service($GLOBALS['HTTP_RAW_POST_DATA']);
269 }
```

query_all_condutor ()

```

124                                     {
125
126     $condutor = new condutor(null,null,null);
127     $result = $condutor->query_all_condutor();
128     if ($result) {
129         return $result;
130     }
131     return "erro select condutor";
132 }
```

query_all_condutor_veiculo ()

```

203                                     {
204
205     $condutor_veiculo = new condutor_veiculo(null,null);
206     $result = $condutor_veiculo->query_all_condutor_veiculo();
207     if ($result) {
208         return $result;
209     }
210     return "erro select query_all_condutor_veiculo";
211 }
```

query_all_localizacao ()

```

243                                     {
244
245     $localizacao = new localizacao(null,null);
246     $result = $localizacao->query_all_localizacao();
247     if ($result) {
```

```

248         return $result;
249     }
250     return "erro select localizacao";
251 }

```

query_all_percurso ()

```

223     {
224
225     $percurso = new percurso(null);
226     $result = $percurso->query_all_percurso();
227     if ($result) {
228         return $result;
229     }
230     return "erro select percurso";
231 }

```

query_all_percurso_ponto ()

```

193     {
194
195     $percurso_ponto = new percurso_ponto(null,null);
196     $result = $percurso_ponto ->query_all_percurso_ponto();
197     if ($result) {
198         return $result;
199     }
200     return "erro select query_all_percurso_ponto ";
201 }

```

query_all_ponto ()

```

213     {
214
215     $ponto = new ponto(null);
216     $result = $ponto->query_all_ponto();
217     if ($result) {
218         return $result;
219     }
220     return "erro select ponto";
221 }

```

query_all_veiculo ()

```

233     {
234
235     $veiculo = new veiculo(null,null,null,null,null,null);
236     $result = $veiculo->query_all_veiculo();
237     if ($result) {
238         return $result;
239     }
240     return "erro select veiculo";
241 }

```

registerMethod (\$nameMethod)

```

262     {
263     $this->server->register($nameMethod);
264 }

```

update_localizacao (\$id, \$lat, \$lon)

```

177     {
178
179     // $percurso_ponto = new percurso_ponto(null,null);
180
181     // $result = $percurso_ponto ->query_all_percurso_ponto();
182     $result=localizacao::update_localizacao($lat,$lon,$id);
183
184     if ($result) {
185         return $result;
186     }
187     return "erro update localizacao ";

```

```
188     }
```

Documentação dos campos e atributos

\$server = NULL

A documentação para esta classe foi gerada a partir do seguinte ficheiro:

- `class.webservice.php`

Documentação do ficheiro

Referência ao ficheiro class.condutor.php

Estruturas de Dados

- class `condutor`

Variáveis

- `if (0 > version_compare(PHP_VERSION, '5'))`

Documentação das variáveis

if

```
Valor inicial:{  
    echo '<h2>Result</h2><pre>'
```

modeloSemTitulo - **class.condutor.php**

\$Id\$

This file is part of modeloSemTitulo.

Automatically generated on 09.12.2015, 09:42:36 with ArgoUML PHP module (last revised

Date

2010-01-12 20:14:42 +0100 (Tue, 12 Jan 2010)

)

Autor:

firstname and lastname of author, author@example.org

Referência ao ficheiro `class.condutor_veiculo.php`

Estruturas de Dados

- `class condutor_veiculo`

Variáveis

- `if (0 > version_compare(PHP_VERSION, '5'))`

Documentação das variáveis

`if(0 > version_compare(PHP_VERSION, '5'))`

modeloSemTitulo - `class.Ponto.php`

`Id`

This file is part of modeloSemTitulo.

Automatically generated on 09.12.2015, 09:42:36 with ArgoUML PHP module (last revised

Date

2010-01-12 20:14:42 +0100 (Tue, 12 Jan 2010)

)

Autor:

firstname and lastname of author, `author@example.org`

Referência ao ficheiro `class.localizacao.php`

Estruturas de Dados

- `class localizacao`

Variáveis

- `if (0 > version_compare(PHP_VERSION, '5'))`

Documentação das variáveis

`if(0 > version_compare(PHP_VERSION, '5'))`

modeloSemTitulo - `class.localizacao.php`

\$Id\$

This file is part of modeloSemTitulo.

Automatically generated on 09.12.2015, 09:42:36 with ArgoUML PHP module (last revised

Date

2010-01-12 20:14:42 +0100 (Tue, 12 Jan 2010)

)

Autor:

firstname and lastname of author, `author@example.org` include `veiculo_localizacao`

firstname and lastname of author, `author@example.org`

Referência ao ficheiro class.percurso.php

Estruturas de Dados

- class **Percurso**

Variáveis

- **if** (0 > version_compare(PHP_VERSION, '5'))
-

Documentação das variáveis

if(0 > version_compare(PHP_VERSION, '5'))

modeloSemTitulo - **class.Ponto.php**

\$Id\$

This file is part of modeloSemTitulo.

Automatically generated on 09.12.2015, 09:42:36 with ArgoUML PHP module (last revised

Date

2010-01-12 20:14:42 +0100 (Tue, 12 Jan 2010)

)

Autor:

firstname and lastname of author, author@example.org

Referência ao ficheiro class.percurso_ponto.php

Estruturas de Dados

- class `percurso_ponto`

Variáveis

- `if (0 > version_compare(PHP_VERSION, '5'))`

Documentação das variáveis

`if(0 > version_compare(PHP_VERSION, '5'))`

modeloSemTitulo - **class.Ponto.php**

\$Id\$

This file is part of modeloSemTitulo.

Automatically generated on 09.12.2015, 09:42:36 with ArgoUML PHP module (last revised

Date

2010-01-12 20:14:42 +0100 (Tue, 12 Jan 2010)

)

Autor:

firstname and lastname of author, author@example.org

Referência ao ficheiro class.ponto.php

Estruturas de Dados

- class **Ponto**

Variáveis

- **if** (0 > version_compare(PHP_VERSION, '5'))

Documentação das variáveis

if(0 > version_compare(PHP_VERSION, '5'))

modeloSemTitulo - **class.Ponto.php**

\$Id\$

This file is part of modeloSemTitulo.

Automatically generated on 09.12.2015, 09:42:36 with ArgoUML PHP module (last revised

Date

2010-01-12 20:14:42 +0100 (Tue, 12 Jan 2010)

)

Autor:

firstname and lastname of author, author@example.org

Referência ao ficheiro class.veiculo.php

Estruturas de Dados

- class **veiculo**

Variáveis

- **if** (0 > version_compare(PHP_VERSION, '5'))

Documentação das variáveis

if(0 > version_compare(PHP_VERSION, '5'))

modeloSemTitulo - **class.veiculo.php**

\$Id\$

This file is part of modeloSemTitulo.

Automatically generated on 09.12.2015, 09:42:36 with ArgoUML PHP module (last revised

Date

2010-01-12 20:14:42 +0100 (Tue, 12 Jan 2010)

)

Autor:

firstname and lastname of author, author@example.org

Referência ao ficheiro class.webservice.php

Estruturas de Dados

- class **WS1_webservice**

Namespaces

- **WS1**

Variáveis

- **if** (0 > version_compare(PHP_VERSION, '5'))
 - **\$ws** = new **WS1_webservice()**
 - **\$namespace** = "http://127.0.0.1:3306/WS1/class.webservice.php"
 - **\$ws server wsdI schemaTargetNamespace** = **\$namespace**
-

Documentação das variáveis

\$namespace = "http://127.0.0.1:3306/WS1/class.webservice.php"

\$ws = new **WS1_webservice()**

if(0 > version_compare(PHP_VERSION, '5'))

include WS1_activity

Autor:

firstname and lastname of author, author@example.org

\$ws server wsdI schemaTargetNamespace = **\$namespace**

Referência ao ficheiro index.php

Variáveis

- **\$client** = new nusoap_client("http://127.0.0.1:80/WS1/class.webservice.php")
 - **\$result** = \$client->call('WS1_webservice.query_all_percurso_ponto', array(null, null))
 - **foreach** (\$result as \$key=> \$value)
-

Documentação das variáveis

\$client = new nusoap_client("http://127.0.0.1:80/WS1/class.webservice.php")

\$result = \$client->call('WS1_webservice.query_all_percurso_ponto', array(null, null))

foreach(\$result as \$key=> \$value)

Referência ao ficheiro interface.php

Variáveis

- **\$client** = new nusoap_client("http://127.0.0.1/WS2/class.webservice.php")
 - **\$origem** = get('origem')
 - **\$destino** = get('destino')
 - **\$err** = \$client->getError()
 - **if(\$err) \$result** = \$client->call('WS2_webservice.direcoes', array('origem' => \$origem, 'destino' => \$destino))
 - **if(\$client->fault) else**
-

Documentação das variáveis

\$client = new nusoap_client("http://127.0.0.1/WS2/class.webservice.php")

\$destino = get('destino')

\$err = \$client->getError()

\$origem = get('origem')

if (\$err) \$result = \$client->call('WS2_webservice.direcoes', array('origem' => \$origem, 'destino' => \$destino))

if (\$err) else

```
Valor inicial:{  
    $err = $client->getError()
```


Referência ao ficheiro interface_condutor.php

Variáveis

- **\$action** = get('action')
 - **\$client** = new nusoap_client("http://127.0.0.1:80/WS1/class.webservice.php")
 - **if(\$action=='novo') else** {\$result = \$client->call('WS1_webservice.query_all_condutor', array(null, null,null))}
-

Documentação das variáveis

\$action = get('action')

\$client = new nusoap_client("http://127.0.0.1:80/WS1/class.webservice.php")

if (\$err) else {\$result = \$client->call('WS1_webservice.query_all_condutor', array(null, null,null))}

Referência ao ficheiro interface_condutor_veiculo.php

Variáveis

- **\$client** = new nusoap_client("http://127.0.0.1:80/WS1/class.webservice.php")
 - **\$id_condutor** = get('id_condutor')
 - **\$id_veiculo** = get('id_veiculo')
 - **\$err** = \$client->getError()
 - **if(\$err) \$result** = \$client->call('WS1_webservice.criar_condutor_veiculo', array('id_condutor' => \$id_condutor,'id_veiculo' => \$id_veiculo))
 - **if(\$client->fault) else**
-

Documentação das variáveis

\$client = new nusoap_client("http://127.0.0.1:80/WS1/class.webservice.php")

\$err = \$client->getError()

\$id_condutor = get('id_condutor')

\$id_veiculo = get('id_veiculo')

if (\$err) \$result = \$client->call('WS1_webservice.criar_condutor_veiculo', array('id_condutor' => \$id_condutor,'id_veiculo' => \$id_veiculo))

if (\$err) else

```
Valor inicial:{  
    $err = $client->getError()
```

Referência ao ficheiro interface_localizacao.php

Variáveis

- **\$action** = get('action')
 - **\$client** = new nusoap_client("http://127.0.0.1:80/WS1/class.webservice.php")
 - **\$id** = get('id_localizacao')
 - **\$latitude** = get('latitude')
 - **\$longitude** = get('longitude')
 - **if(\$action=='novo') else**
-

Documentação das variáveis

\$action = get('action')

\$client = new nusoap_client("http://127.0.0.1:80/WS1/class.webservice.php")

\$id = get('id_localizacao')

\$latitude = get('latitude')

\$longitude = get('longitude')

if (\$err) else

```
Valor inicial:{  
    $result = $client->call('WS1_webservice.update_localizacao', array('id' =>  
$id, 'latitude' => $latitude, 'longitude' => $longitude))
```

Referência ao ficheiro interface_percurso.php

Variáveis

- **\$action** = get('action')
 - **\$client** = new nusoap_client("http://127.0.0.1:80/WS1/class.webservice.php")
 - **if(\$action=='novo') else**
-

Documentação das variáveis

\$action = get('action')

\$client = new nusoap_client("http://127.0.0.1:80/WS1/class.webservice.php")

if (\$err) else

```
Valor inicial:{  
$result = $client->call('WS1_webservice.query_all_percurso', array(null))
```

Referência ao ficheiro interface_percurso_ponto.php

Variáveis

- **\$client** = new nusoap_client("http://127.0.0.1:80/WS1/class.webservice.php")
 - **\$id_percurso** = get('id_percurso')
 - **\$id_ponto** = get('id_ponto')
 - **\$err** = \$client->getError()
 - **if(\$err) \$result** = \$client->call('WS1_webservice.criar_percurso_ponto', array('id_percurso' => \$id_percurso, 'id_ponto' => \$id_ponto))
 - **if(\$client->fault) else**
-

Documentação das variáveis

\$client = new nusoap_client("http://127.0.0.1:80/WS1/class.webservice.php")

\$err = \$client->getError()

\$id_percurso = get('id_percurso')

\$id_ponto = get('id_ponto')

if (\$err) \$result = \$client->call('WS1_webservice.criar_percurso_ponto', array('id_percurso' => \$id_percurso, 'id_ponto' => \$id_ponto))

if (\$err) else

```
Valor inicial:{  
    $err = $client->getError()
```

Referência ao ficheiro interface_ponto.php

Variáveis

- **\$action** = get('action')
 - **\$client** = new nusoap_client("http://127.0.0.1:80/WS1/class.webservice.php")
 - **if(\$action=='novo') else**
-

Documentação das variáveis

\$action = get('action')

\$client = new nusoap_client("http://127.0.0.1:80/WS1/class.webservice.php")

if (\$err) else

```
Valor inicial:{  
$result = $client->call('WS1_webservice.query_all_ponto', array(null))
```

Referência ao ficheiro interface_veiculo.php

Variáveis

- **\$action** = get('action')
 - **\$client** = new nusoap_client("http://127.0.0.1:80/WS1/class.webservice.php")
 - **if(\$action=='novo') else**
-

Documentação das variáveis

\$action = get('action')

\$client = new nusoap_client("http://127.0.0.1:80/WS1/class.webservice.php")

if (\$err) else

```
Valor inicial:{  
$result = $client->call('WS1_webservice.query_all_veiculo', array(null, null, null, null,  
null,null))
```

Índice

INDEX